ホスト協調によるSSHパスワード総当たり攻撃の防御

阿波連　良尚[†]　新城　靖[††]　佐藤　聡[††]
中井　央[†††]　板野　肯三[††]

1. はじめに

インターネットがインフラとして重要な役割を担っている現代社会において，不正アクセスや迷惑メール送信などに悪用されるボットネットの存在は大きな脅威となっている．ボットネットとは，ボットと呼ばれる種類のマルウェアに感染したホストで構成されるネットワークである．本研究では，ボットネットが行うSecure Shell(SSH)へのパスワード総当たり攻撃に着目する．SSHへのパスワード総当たり攻撃を行っているホストは，ボットに感染している可能性がある[5]．ボットに感染したホストは，IRCなどを通じて攻撃者の指示を受け，スパムメールの送信やDDoS攻撃などに悪用されることがある[1]．このため，SSHへの攻撃を検出して攻撃元をブロックすることは，自組織内の計算機を保護するために有効であると考えられる．

そこで，本研究ではローカルネットワーク内で攻撃記録の共有と解析を行い，パスワード総当たり攻撃を行っていると判断されたIPアドレスを遮断する手法を提案する．また，ホスト間で協調して対策を行う必要性について検証する．

2. 既存手法とその問題点

SSHは，遠隔地にある計算機にアクセスするためのプロトコルとして広く利用されている．SSHを安全に利用するには，公開鍵による認証を行うことが推奨される．ところが，大学の計算機環境のように，多数の計算機を多数のユーザが利用する環境では，管理上の都合からパスワードによる認証が利用されることが多い[3]．しかし，パスワード認証は総当たり攻撃によって突破することが可能であり，SSHをターゲットとした総当たり攻撃は広く行われている[4]．

SSHへのパスワード総当たり攻撃を検出する方法として，セキュリティログを解析して動的にファイアウォールルールを構成する手法が存在する[3]．これを発展させて，セキュリティログをsyslogを用いてネットワーク全体で共有することにより，中央で解析を行ってホスト間で解析結果を共有する手法も提案されている[2]．しかし，セキュリティログ全体を転送する手法では，解析サーバに秘匿すべき内容まで送信してしまう．このため，たとえば組織内の管理区分を越えての対策を行うには問題がある．

一方，各ホストでログの解析を行い，ブロックすべきIPアドレスをホスト間で共有する手法も提案されている[6]．この手法では，セキュリティログの秘匿すべき内容がネットワークに流れることはないが，後述する分散型の攻撃に対応することが困難である．

そこで，各ホストではログの解析のうち攻撃元IPアドレスの抽出のみを行い，ローカルネットワーク内で記録の統合・解析を行って，解析結果をホスト間で共有する．

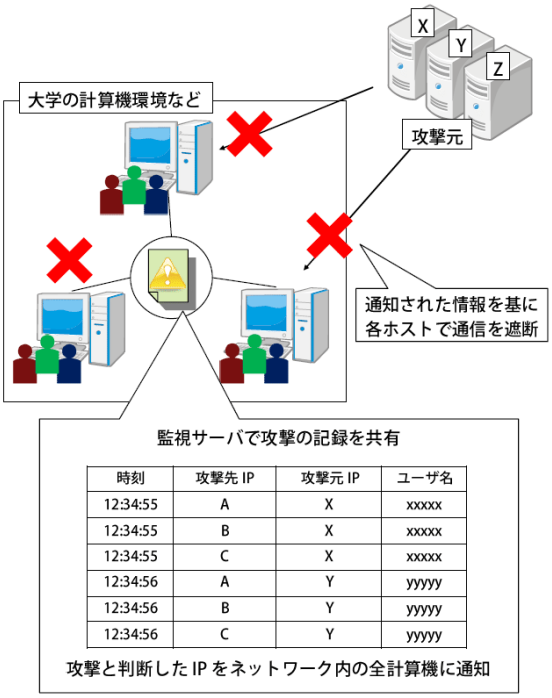

図1　提案手法の概略図

3. ホスト協調による攻撃の防御

本手法の概略を**図1**に示す．各ホストにセキュリティログを監視する**デーモン**を稼働させる．このデーモンは，ログを監視し，SSHのパスワード認証失敗の記録

† 筑波大学第三学群情報学類
†† 筑波大学システム情報工学研究科
††† 筑波大学図書館情報メディア研究科

を抽出する．ここで，パスワード認証失敗とは，以下の二つのケースを指す．

- 認証を要求したユーザがシステムに存在しない
- ユーザは存在するが，パスワードが間違っている

認証失敗を検出したデーモンは，**解析サーバ**に { 日時，攻撃元 IP アドレス，攻撃先 IP アドレス，ユーザ名 } を送信する．解析サーバは，各ホストから受信した認証失敗記録に基づいて，同一の IP アドレスから 10 分間に 10 回以上の認証失敗があった場合にパスワード総当たり攻撃と判断する．ブロックされるべきと判断された IP アドレスの情報は，解析サーバから各ホストに配布される．各ホストでは，iptables や ipfw のようなファイアウォールによって，指定された IP アドレスとの全通信を遮断する．通信遮断の期間は 10 分間とし，at コマンドなどのジョブ管理ツールを用いて解除を行う．

4. ホスト協調の必要性

SSH へのパスワード総当たり攻撃の現状とその傾向について調査するため，2009 年 5 月より現在までハニーポットを設置している．認証要求を全て拒否するよう設定した SSH サーバのみを稼働させた計算機に，16 個の連続したグローバル IP アドレスを割り振って外部から接続可能にした．他にサービスは公開せず，DNS にも登録していないことから，このホストに対する SSH 接続はパスワード総当たり攻撃であると考えられる．

4.1 SSH パスワード総当たり攻撃の傾向

SSH パスワード総当たり攻撃の特徴として，以下が挙げられる[3]．

- 攻撃には，root や admin など通常の利用者が使用しないユーザ名が用いられる
- ツールプログラムなどを使用して，連続してログインを試みる
- ポートスキャンなどによって攻撃対象を見つけ，手当たり次第にログインを試みる

これらの特徴に加えて，設置したハニーポットで収集した記録から分散型の攻撃と思われる特徴を発見した．これは，それぞれ 1 分〜3 分程度の時間間隔を置いて，292 個の異なる IP アドレスから攻撃を受けたケースである．攻撃の継続時間はおよそ 21 時間で，1 攻撃元 IP アドレスあたり 1 回〜10 回，合計すると 961 回のログイン試行を受けた．このとき，攻撃対象に隣接する IP アドレスを持つホストに対しても同様の記録が残されていた．複数のホストに対して，ほぼ同時期に，存在しないユーザ名でのログインを試みることは，通常の利用では考えづらい．よって，これは分散型のパスワード総当たり攻撃である可能性が高いと判断できる．

4.2 分散型攻撃の検出

前節で述べたような分散型の攻撃については，スタンドアロンでの検出が難しい．これは，特定の IP アドレスからのログイン試行の回数が少なく，一般利用者の誤操作などによるログイン失敗と区別がつかないためである．ここで，前節で述べたように隣接する IP アドレスに対しても攻撃を行うケースでは，ホスト間での情報共有により検出を行うことができると考えられる．これにより，分散型攻撃についても対策を行える．

5. ま と め

本稿では，SSH に対するパスワード総当たり攻撃の現状と既存の対策手法について述べ，改善した対策手法を提案した．パスワード認証を利用せざるを得ない環境において，総当たり攻撃への対処は重要な課題であるといえる．本稿で述べた方法により，利用者の利便性を損ねることなく，SSH パスワード総当たり攻撃に対して精度の高い検出と防御を行うことができる．

スタンドアロンの検出システムについては，Ruby での実装を完了し，RedHat Enterprise Linux 4 および Mac OS X 10.4 での動作を確認した．通信遮断については，RedHat Enterprise Linux では iptables を，Mac OS X では ipfw を利用している．ホスト協調システムについては，dRuby を用いた実装を進めている．攻撃の判別やブロック期間などのパラメータについては，攻撃開始から検出までの時間を可能な限り短くするため，今後チューニングを行う予定である．

参 考 文 献

1) Berthier, R. G.: Advanced Honeypot Architecture for Network Threats Quantification, PhD Thesis, University of Maryland (2009).
2) 大隅淑弘，山井成良：ホスト間連携を可能にするパスワード総当たり攻撃対策手法，情報処理学会研究報告， Vol. 2007, No. 93, pp. 49–54 (2007).
3) 大隅淑弘，山井成良，井上一郎二：アクセス制御ファイルの動的変更による SSH 総当たり攻撃への対策，学術情報処理研究， No. 11, pp. 68–73 (2007).
4) Ramsbrock, D., Berthier, R. and Cukier, M.: Profiling Attacker Behavior Following SSH Compromises, *Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, Washington, DC, USA, IEEE Computer Society, pp. 119–124 (2007).
5) Seifert, C.: Analyzing Malicious SSH Login Attempts, `http://www.securityfocus.com/infocus/1876` (2006). 2009 年 10 月 29 日閲覧.
6) Thames, J.L., Abler, R. and Keeling, D.: A Distributed Active Response Architecture for Preventing SSH Dictionary Attacks, *Proceedings of IEEE SouthEastCon*, Huntsville, AL, USA, pp. 84–89 (2008).